Fundamentos de Algoritmia

Grados en Ingeniera Informática. Grupos C, F y DG

Examen Convocatoria Ordinaria, 3 de Febrero de 2021.

Nombre: _____ **Grupo:** _____

Laboratorio: _____ **Puesto:** _____ **Usuario de DOMjudge:** _____

Normas de realizacin del examen

1. Debes programar soluciones para cada uno de los tres ejercicios, probarlas y entregarlas en el juez automtico accesible en la direccin `http://exacrc/domjudge/team`.
2. Escribe comentarios que expliquen tu solucin, justifiquen por qu se ha hecho as y ayuden a entenderla. Calcula la complejidad de todas las funciones que implementes.
3. En el juez te identificars con el nombre de usuario y contrasea que has recibido al comienzo del examen. El nombre de usuario y contrasea que has estado utilizando durante la evaluacin continua **no** son vlidos.
4. Escribe tu **nombre y apellidos** en un comentario en la primera lnea de cada fichero que subas al juez.
5. Tus soluciones sern evaluadas por el profesor independientemente del veredicto del juez automtico. Para ello, el profesor tendr en cuenta **exclusivamente** el ltimo envo que hayas realizado de cada ejercicio.

1. **(3.75 puntos)** Dado un vector v de $n \geq 0$ enteros positivos y un natural $k \geq 0$, se desea encontrar la longitud del segmento ms largo que no contenga ningn subsegmento con (estrictamente) ms de k nmeros pares.

    1. (0.25 puntos) Define un predicado $todosPares(v, p, q)$ que devuelva cierto si y solo si todos los elementos del vector v contenidos entre las posiciones p (incluida) y q (excluida) son pares (supuesto $0 \leq p \leq q \leq v.size()$).
    2. (0.25 puntos) Utilizando *todosPares* define un predicado $noMas(v, p, q, k)$ que devuelva cierto si y solo si todos los posibles segmentos del vector v entre las posiciones p y q (supuesto $0 \leq p \leq q \leq v.size()$) que tengan todos sus elementos pares tienen una longitud menor o igual que k.
    3. (0.25 puntos) Utilizando el predicado *noMas*, especifica una funcin que dado un vector de enteros positivos de longitud ≥ 0 y un natural $k \geq 0$, devuelva la longitud del segmento ms largo que no contiene ningn subsegmento con ms de k nmeros pares.
    4. (2.5 puntos) Disea e implementa un algoritmo iterativo que resuelva el problema propuesto.
    5. (0.5 puntos) Escribe el invariante del bucle que permite demostrar la correccin del mismo y proporciona una funcin de cota.
    6. (0.25 puntos) Indica el coste asinttico del algoritmo en el caso peor y justifica adecuadamente tu respuesta.

Entrada

La entrada comienza con una lnea que contiene el nmero de casos de prueba. Cada caso de prueba contendr el valor del nmero de elementos n, el valor de k y a continuacin los elementos de la secuencia.

Salida

Por cada caso de prueba el programa escribir una lnea con la longitud del segmento ms largo solicitado en el enunciado.

Entrada de ejemplo

```
5
8 4
1 2 6 4 8 10 3 7
8 2
1 2 4 3 9 6 8 7
4 7
2 4 6 8
3 0
2 4 6
3 1
2 4 6
```

Salida de ejemplo

```
6
8
4
0
1
```

2.(2.5 puntos) Un vector de enteros positivos es *extrao* si la suma de los nmeros pares de la primera mitad ms el producto de los nmeros impares de la primera mitad es menor o igual que el producto de los nmeros pares de la segunda mitad ms la suma de los nmeros impares de la segunda mitad, y al menos una de sus dos mitades es extraa. Un vector vaco o con un nico elemento siempre es extrao.

Se pide:

1. (1.75 puntos) Escribe un algoritmo recursivo eficiente que permita resolver el problema para un vector dado suponiendo que el nmero de elementos es una potencia de dos.
2. (0.75 puntos) Escribe la recurrencia que corresponde al coste de la funcin recursiva indicando claramente cual es el tamao del problema. Indica tambin a qu orden de complejidad asinttica pertenece dicho coste.

Entrada

La entrada comienza con una lnea que contiene el nmero de casos de prueba. Cada caso de prueba contendr el nmero de elementos del vector y a continuacin una lnea con los elementos.

Salida

Por cada caso de prueba el programa escribir `NO` si el vector no es extrao segn la definicin o `SI` si el vector lo es.

Entrada de ejemplo

```
5
2
1 2
2
2 2
4
1 2 2 2
4
2 4 1 2
4
2 2 2 2
```

Salida de ejemplo

```
SI
NO
SI
NO
NO
```

3. (4 puntos) La pandemia Covid-19 ha forzado a muchos restaurantes a digitalizarse. El restaurante Come Sano dispone de n plazas y conoce las distancias d_{ij} entre cada dos plazas i y j, $0 \leq i, j \leq n-1$. Tiene reservas para $m \leq n$ comensales y una matriz de booleanos c_{kl} le indica si dos comensales k y l, $0 \leq k, l \leq m-1$ son o no allegados. Dos personas allegadas pueden sentarse a cualquier distancia, pero dos no allegados han de estar separados al menos dos metros entre s. Se quiere decidir cmo sentar a todos los comensales en las plazas del restaurante de forma que se respeten las distancias de seguridad y se maximice el nmero de parejas de comensales allegados donde los dos componentes estn sentados a (estrictamente) menos de dos metros de distancia.

Notas: se entiende que la pareja formada por el comensal 1 y el comensal 3 es la misma pareja que la formada por el 3 y el 1, as que solo se cuenta una vez. Las matrices de distancias y allegados son simtricas y con 0s en la diagonal.

- (3 puntos) Implementa un algoritmo de vuelta atrs que resuelva el problema. Explica claramente los marcadores que has utilizado.
- (1 puntos) Plantea al menos una funcin de poda de optimalidad e implemntala en tu algoritmo.

Entrada

La entrada comienza con una lnea que contiene el nmero de casos de prueba. Cada caso de prueba contendr inicialmente el valor del nmero de plazas del restaurante n, de comensales m. A continuacin n filas con las distancias d_{ij} entre las plazas del restaurante. Y finalmente m filas que indican mediante 0s (falso) y 1s (cierto) si dos comensales son o no allegados.

Salida

Por cada caso de prueba el programa escribir CANCELA si no se puede sentar a los comensales respetando las distancias de seguridad, y en caso contrario escribir PAREJAS seguida de la cantidad mxima de parejas de comensales allegados sentados a menos de dos metros de distancia.

Entrada de ejemplo

```
3
5 4
0 1 1 4 5
1 0 1 3 4
1 1 0 2 3
4 3 2 0 1
5 4 3 1 0
0 1 1 0
1 0 1 0
1 1 0 0
0 0 0 0
2 2
0 0.5
0.5 0
0 0
0 0
4 4
0 0.5 3.5 4
0.5 0 2.5 3
3.5 2.5 0 3
4 3 3 0
0 1 1 0
1 0 1 0
1 1 0 0
0 0 0 0
```

Salida de ejemplo

```
PAREJAS 3
CANCELA
PAREJAS 1
```